HICKMAN PALERMO TRUONG & BECKER LLP
2055 GATEWAY PLACE, SUITE 550
SAN JOSE, CALIFORNIA 95110-1089
TEL: (408) 414-1080
FAX: (408) 414-1076

**FACSIMILE TRANSMITTAL SHEET**

| TO: | FROM: |
|---|---|
| Examiner William S. Powers | Craig G. Holmes |
| COMPANY: U.S.P.T.O. | DATE: APRIL 25, 2006 |
| FAX NUMBER: 571 273 8573 | TOTAL NO. OF PAGES INCLUDING COVER: [redacted] 9 |
| PHONE NUMBER: 571-272-8573 | SENDER'S REFERENCE NUMBER: 50325-0598 |
| RE: Interview Request | YOUR REFERENCE NUMBER: 10/040,050 |

☐ URGENT ☑ FOR REVIEW ☐ PLEASE COMMENT ☐ PLEASE REPLY ☐ PLEASE RECYCLE

Examiner Powers,

As you requested in the phone message that you left today following our earlier phone conversation, please find attached the Applicant Initiated Interview Request Form with a proposed date/time of **Friday, April 28, 2006, at 2:00 PM EST** (11:00 AM PST).

If the Examiner grants this request for an Interview, but the proposed date and time are not acceptable, the Applicant respectfully requests that the Examiner propose an alternate date and time, keeping in mind that the Applicant is located in California and the 3 hour earlier time difference resulting therefrom. Thus, the Applicant would prefer to conduct the Interview during the later morning or afternoon of the Examiner's time.

Also, as you requested, please find the attached discussion of arguments that the Applicant would like to discuss during the Interview. Please note that the Applicant is not proposing any claim amendments at this time and rather wishes to focus the interview on the basis of the rejections provide in the Final Office Action.

Respectfully submitted,

Craig Holmes
Reg. No. 44,770

PTOL-413A (09-04)
Approved for use through 07/31/2006. OMB 0651-0031
U.S. Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE

# Applicant Initiated Interview Request Form

Application No.: 10/040,050 First Named Applicant: Mahesh S. Maddury
Examiner: William S. Powers Art Unit: 2134 Status of Application: Final OA mailed 3/1/06

**Tentative Participants:**
(1) Ex. Powers (2) Craig Holmes
(3) _____ (4) _____

**Proposed Date of Interview:** Friday, April 28, 2006 **Proposed Time:** 2:00 ~~(AM/PM)~~ EST (11:00 AM PST)

**Type of Interview Requested:**
(1) [X] Telephonic (2) [ ] Personal (3) [ ] Video Conference

**Exhibit To Be Shown or Demonstrated:** [ ] YES [X] NO
**If yes, provide brief description:** _____

## Issues To Be Discussed

| Issues (Rej., Obj., etc) | Claims/ Fig. #s | Prior Art | Discussed | Agreed | Not Agreed |
|---|---|---|---|---|---|
| (1) 102(b) Rej. | Claim 5 | Naccache | [ ] | [ ] | [ ] |
| (2) 103(a) Rej. | Claims 1, 2, 3, 4 12, 14, 15, 16 | Naccache + Admitted prior Art. | [ ] | [ ] | [ ] |
| (3) _____ | _____ | _____ | [ ] | [ ] | [ ] |
| (4) _____ | _____ | _____ | [ ] | [ ] | [ ] |

[ ] Continuation Sheet Attached

**Brief Description of Arguments to be Presented:**

See attached sheet.

**An interview was conducted on the above-identified application on** _____.
**NOTE: This form should be completed by applicant and submitted to the examiner in advance of the interview (see MPEP § 713.01).**
**This application will not be delayed from issue because of applicant's failure to submit a written record of this interview. Therefore, applicant is advised to file a statement of the substance of this interview (37 CFR 1.133(b)) as soon as possible.**

[signature] _____
Applicant/Applicant's Representative Signature Examiner/SPE Signature

Craig G. Holmes
Typed/Printed Name of Applicant or Representative

44,770
Registration Number, if applicable

## Attachment for "Applicant Initiated Interview Request Form"

**Description of Arguments to be Presented:**

The following description of the arguments is preceded by an example embodiment of Claim 1, as previously provided in the response to the previous Office Action, which the Examiner may find helpful in understanding the later presented issues. Following the example embodiment of Claim 1, the following three issues are addressed: (1) the feature of a prime modulus (as included in all the claims), (2) the feature of modular exponentiation (as included in all the claims), and (3) the use of a power value that is equal to two less than the prime modulus (as included in Claims 1-4 and 12-16) being rejected upon Euler's Theorem as "admitted prior art."

### Example Embodiment of Claim 1

The features of Claim 1 correspond to an implementation of expression (23) in the application, namely $a^{p-2} = a^{-1} \bmod p$, in which a modulo multiplicative inverse, $a^{-1}$ mod p, is determined based on modulo exponentiation, $a^{p-2}$ mod p, with p being chosen to be a **prime modulus**. The approach of Claim 1 is implemented using a **modulo exponentiation** block, which avoids the problems of using the extended Euclidean algorithm (EEA) that is an iterative approach and slow for large numbers, which is today common with key sizes of 1024, 2048, or even more bits. As explained in the Application, EEA is typically implemented as a multiplicative inverse (MI) block through application specific integrated circuits (ASICs), that occupy a large area of chip "real estate." (Application, pages 3-4.)

However, in the approach of Claim 1, existing blocks, such as a **modulo exponentiation** (ME) block, can be used that have smaller area requirements when implemented on a chip. This improvement is at the "expense" of requiring that the modulus be a prime modulus, which is required in deriving Equation (23) as illustrated in the Application, although such an expense is generally outweighed by the result of being able to calculate a multiplicative inverse using modular exponentiation in lieu of an ASIC that implements the larger and more time consuming MI circuitry.

Specifically in Claim 1, the **modulo exponentiator block** is used in "determining a multiplicative inverse of the first integer data value modulo a prime modulus by computing a first quantity modulo the **prime modulus** data value." For example, the first integer data value is the value for which the multiplicative inverse is desired, such as "a" in expression (23) of the application. The prime modulus is "p" in expression (23).

Next in Claim 1, the "first quantity equals, modulo the **prime modulus** data value, the first integer data value raised to a **power of a second quantity**." For example, the first quantity is "a" in expression (23) modulo the prime modulus "p" raised to the power of the second quantity.

Then in Claim 1, the "**second quantity is two less than the prime modulus data value.**" For example, the second quantity is the exponent of expression (23), namely "**p-2**" or two less than the prime modulus.

### Issue 1 – Use of a Prime Modulus

All of the claims feature the use of a prime modulus, yet the Applicant is unable to find where a prime modulus is mentioned in the cited portions of *Naccache*, although other portions of Naccache do refer to a prime modulus. As the Final Office Action currently stands, it does not appear to the Applicant that prima facie rejections have been established with respect to the "prime modulus" feature of the claim since none of the cited portions of Naccache disclose a prime modulus. Therefore, the Applicant would like to clarify with the Examiner during the Interview what the basis is for the "prime modulus" feature of the claims to ensure that this feature has not been overlooked in establishing the rejections of the Final Office Action.

For example, the rejections of Claims 1, 5, 14 and 15 cites Col. 4, lines 45-48 and Figure 2 with respect to the portion of the claims referring to a prime modulus, yet neither Col. 4, lines 45-48 nor Figure 2 discloses a prime modulus. In fact, in that cited portions of Naccache, there is only a reference to using a modulus "n" that is not described as being a prime modulus.

While the Applicant notes that other portions of Naccache refer to the use of a prime modulus "p" in the DSA scheme (see Col. 5, lines 41-42, for example), the Final Office Action's rejections do not cite those portions of Naccache referring to the prime modulus "p." Thus, the Applicant would like to discuss the basis of the Final Office Action's rejection with respect to the prime modulus feature of the claims, and specifically which portion(s) of Naccache is being relied upon as disclosing the "prime modulus" feature of the claims.

**Issue 2 – Use of a Modular Exponentiation Block/Function**

All of the claims feature modular exponentiation, such as by using a "modulo exponentiation block" or a "modulo exponentiation function." Yet the Final Office Action appears to only cite to Figure 2 of Naccache for this feature of the claims. Yet Figure 2 only discloses a CPU 30 that is illustrated as including "programs or computational resources corresponding to or implementation...exponentiation." (Figure 2; Col. 4, lines 25 – 34). The same is true with respect to Figure 1 of Naccache and the discussion of CPU 11 illustrated thereon as having "modular reduction" but only "exponentiation" (see Col. 3, lines 52-60).

While Figures 1 and 2 and the above cited portions of Naccache refer to "modular reduction" and Figure 2 and its description also referring to "modular inversion," there is no illustration, description, or reference to "modular exponentiation" anywhere within Naccache that the Applicant has been able to locate by either reading and reviewing the reference or performing electronic searches therein. Rather, as far as the Applicant has been able to determine, Naccache only refers to "exponentiation" without characterizing it as being modular. And as discussed in the previous Office Action response, a circuit for performing exponentiation is not the same as a circuit that performs modular exponentiation.

Thus, given that Naccache expressly refers to "modular inversion" and "modular reduction," yet only refers to "exponentiation," the Applicant fails to see any disclosure within Naccache of either a "modulo exponentiation block" or a "modulo exponentiation function" as featured in the Claims. Therefore, the Applicant would like to discuss with the Examiner the basis within Naccache for the Final Office Action's rejection with respect to the "modulo exponentiation block/function" as featured in the claims.

**Issue 3 – Characterizing Equation (23) of the Application as "Euler's Theorem" & Applicant Admitted Prior art**

Claims 1-4 and 12-16 feature the use of an exponent that is equal to the value of "the prime modulus less two," a value "two less than the prime modulus" value, or similar variations thereof. The Final Office Action rejects this feature of Claims 1-4 and 12-16 based on the alleged "Applicant admitted prior art," namely Euler's Theorem as presented in paragraph 43 of the Specification of the Applicant's disclosure. However, for the reasons outlined below, the Applicant respectfully submits that characterizing the entire content of paragraph 43 as Euler's Theorem is incorrect.

In paragraph 43 of the Applicant's specification, a derivation of Equation 23 is presented. The derivation begins with a statement of Euler's Theorem, namely that for two positive numbers "a" and "b" that are *relatively prime* (*e.g.*, their greatest common denominator is 1, which is often expressed as gcd(a,b)=1), the following relation is true: $a^{\phi(b)} = 1 \bmod b$. This equation-based representation of Euler's Theorem is identified in the specification as Equation (21).

Note that "relatively prime" means that the only common denominator between two numbers is 1, but that this does not mean that either number is prime itself. For example, the numbers "3" and "4" are relatively prime because the greatest common denominator between "3" (with denominators of 1 and 3) and "4" (with denominators 1, 2, and 2) is "1." Yet "4" is clearly not prime as the factors of 4 are 1, 2, and 2, with "2" being repeated, meaning that "4" is not a prime number (e.g., a number that is only divisible by "1" and itself).

Thus, the Applicant would agree that Equation (21) of the specification may be characterized as Euler's Theorem. This is consistent with other descriptions of Euler's Theorem, as indicated in the attached examples, and no doubt in other examples that could be found in performing a search of "Euler's Theorem." Note that in the presentation of Euler's Theorem in the specification, which is consistent with the examples provided, the only restrictions on "a" and "b" is that both be a positive number and that "a" and "b" are relatively prime with respect to each other.

The derivation of paragraph 43 then proceeds from Euler's Theorem, represented by Equation (21), as the starting point to reach Equation 22, which is a manipulation of Equation (21) to determine an expression that relies upon modular exponentiation. Note that the use of modular exponentiation in Equation (22) is unlike the statement of Euler's Theorem in Equation (21) that does not involve modular exponentiation. Thus, the Applicant disagrees with the Final Office Action's characterization of Equation (22) as being part of "Euler's Theorem" since the inclusion of modular exponentiation in Equation (22) is different than in Equation (21) that lacks any modular exponentiation.

Then in finally deriving the expression identified as Equation (23) in the specification, the derivation assumes that "b" is equal to a positive prime number. It is only with this further limitation/restriction that the function $\phi(b)$ is equal to "p-1", and thus "$\phi(b)-1$" becomes "p-2" as shown in Equation (23). As a result of the expression of Equation (23), it can be observed that a modular multiplicative inverse can be obtained through the use of modular exponentiation, which is not the case with Euler's Theorem in Equation (21).

Therefore, Equation (23) is clearly and unambiguously different than Euler's Theorem, as represented by Equation (21), because Euler's Theorem only requires "b" be a positive number, whereas Equation (23) requires that "p" be a positive prime number and be relatively prime with respect to "a." As a result, the Applicant respectfully disagrees with the Final Office Action's characterization that all of paragraph 43 of the specification is Euler's Theorem and therefore "admitted prior art." Rather, the Applicant respectfully submits that only Euler's Theorem, as represented by Equation (21), is properly characterized as prior art with the remainder of paragraph 43, including both Equations (22) and (23), not being prior art.

In fact, the rejection of Claims 1-4 and 12-16 on the basis of Equation (23) effectively is a rejection of the Applicant's claims on the Applicant's own invention, as represented by Equation (23), which the Applicant respectfully submits is improper.

**Conclusion**

The Applicant respectfully requests that the Applicant's request for an Examiner Interview be granted so that the basis of the rejections of the claims with respect to the "prime modulus" and "modulo exponentiation" features of the claims can be explained to the Applicant. In particular, with regards to the "modulo exponentiation" feature of the claims, the Applicant is unable to find anything in Naccache that discloses this feature, and the Applicant would appreciate the Examiner providing a description of what is being relied upon as disclosing modular exponentiation.

Also, the Applicant believes that discussing the alleged "Applicant admitted prior art" of Euler's Theorem as presented in Equation (21) of paragraph 43 of the specification would be beneficial to the Examiner in understanding that while Equation (21) may be characterized as "Euler's Theorem," Equation (23) is not properly characterized as "Euler's Theorem" since Equation (23) requires a restriction to a prime modulus that is not part of Euler's Theorem.

US005742534A

# United States Patent [19]

Monier

[11] Patent Number: 5,742,534

[45] Date of Patent: Apr. 21, 1998

[54] **ELECTRONIC CIRCUIT FOR MODULAR COMPUTATION IN A FINITE FIELD**

[75] Inventor: Guy Monier, Rognac, France

[73] Assignee: SGS-Thomson Microelectronics, S.A., Gentilly, France

[21] Appl. No.: 531,952

[22] Filed: Sep. 21, 1995

[30] Foreign Application Priority Data

Sep. 21, 1994 [FR] France ..... 94 11420

[51] Int. Cl.[6] ..... G06F 7/72

[52] U.S. Cl. ..... 364/746.1

[58] Field of Search ..... 364/146.1, 754, 364/757

[56] References Cited

U.S. PATENT DOCUMENTS

5,513,133 4/1996 Cressal et al. ..... 364/754
5,535,225 7/1996 Mayhew et al. ..... 364/746.1
5,602,767 2/1997 Fenwels et al. ..... 364/746.1

FOREIGN PATENT DOCUMENTS

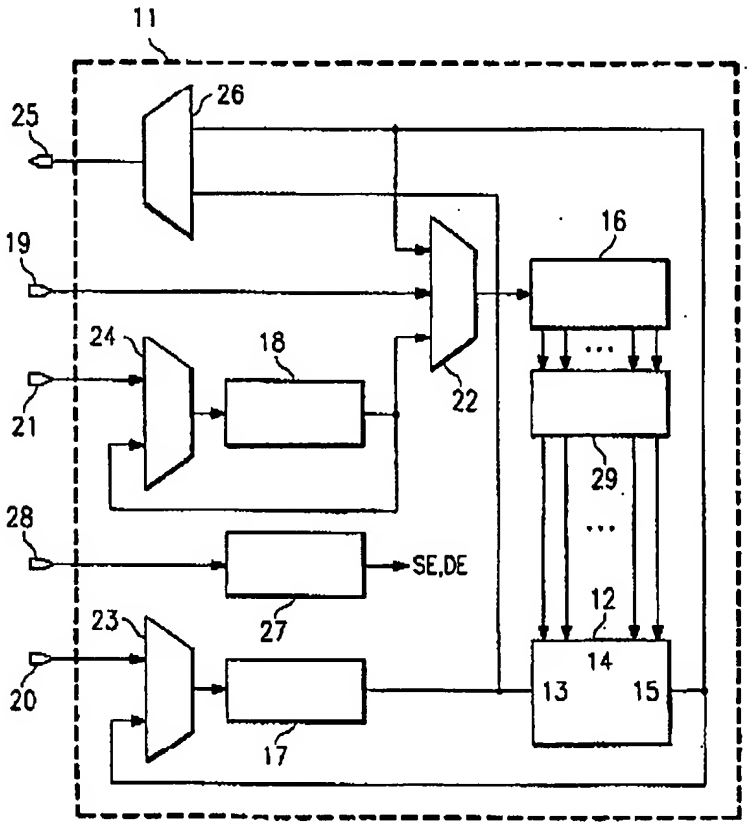
0145533 6/1985 European Pat. Off. .
0531158 3/1993 European Pat. Off. .
0601907 6/1994 European Pat. Off. .

*Primary Examiner*—David H. Malzahn
*Attorney, Agent, or Firm*—Robert Groover; Betty Formby; Matthew Anderson

[57] ABSTRACT

An electronic computation circuit comprises a multiplication operator with a serial input, a parallel input and a serial output, a first register connected by its output to the parallel input of the operator, a second register connected by its output to the serial input of the operator, a third register and a multiplexing circuit to selectively connect at least one data input terminal and the output of the operator to the inputs of the first, second and third registers, and to produce the output of the electronic multiplication circuit. Application to the operations of multiplication, squaring, exponentiation and modular inversion on a finite field denoted GF(2ⁿ).

34 Claims, 3 Drawing Sheets



$Y=X^*Y$

$i=i-1$.

It can thus be seen that:

if $e_i=0$, the algorithm implements a squaring operation,

if $e_i=1$, the algorithm implements a squaring operation followed by a multiplication.

To perform the computation of Y, it is enough to carry out the following procedure:

1) the selection of the inputs of the multiplexers 22, 23 and 24 so as to connect the inputs of the registers 16, 17 and 18 to the data input terminals 19, 20 and 21,

2) loading by the shifting of X into the first, second and third registers 16, 17 and 18,

for i from l−2 to 0:

if $e_i=0$ then:

3) the selection of the multiplexers 22 and 23 so as to connect the inputs of the first and second registers to the output of the multiplication operator,

4) the shifting of the contents of the second register into this register so as to successively give the n bits on which its contents are encoded to the multiplication operator, the n bits that gradually come out of this operator and that represent a squaring operation being simultaneously loaded into the first and second registers (at least so long as i>0)

if $e_i=1$ then:

3) the selection of the multiplexers 22 and 23 so as to connect, firstly, the inputs of the first and third register to the output of the third register and, secondly, the input of the second register to the output of the multiplication operator,

4) the shifting of the contents of the second register into this register so as to successively give the n bits on which its contents are encoded to the multiplication operator, the n bits that gradually come out of this operator and represent the result of a squaring operation being simultaneously loaded into the second register (at least so long as i>0) and, at the same time, the shifting of the contents of the third register into this register (by the looping of the output to the input) and into the first register,

5) the selection of the multiplexers 22 and 23 so as to connect the inputs of the first and second registers to the output of the multiplication operator,

6) the shifting of the contents of the second register into this register so as to successively give the n bits on which its contents are encoded to the multiplication operator, the n bits that gradually come out of this operator and represent the result of a squaring operation being simultaneously loaded into the first and second registers 16 and 17 (at least so long as i>0).

Steps 1) and 2) require n+1 cycles to be performed.

The steps 3) and 4) on the one hand and 5) and 6) on the other hand also require n+1 cycles to be performed.

On the whole, the duration of the processing will be m cycles with:

$$m=(n+1)+[(l-1)+(k-1)]*(n+1), \text{ that is } m=(1+k-1)*(n+1).$$

If i=0, the last step will reveal the desired result at output of the multiplication operator.

The utility of the storage circuit 29 appears clearly in the context of exponentiation. Indeed, during the implementation of the above-described method, a serial loading is carried out, in the first register, of the bits during the



performance of the multiplication and squaring operations, and there must therefore be a means available to hold the states present at the parallel input of the operator in a stable state during the performance of these operations.

Modular Inversion

The inversion amounts to an exponentiation processed identically to that described here above. Indeed, if we consider Euler's theorem, if the greatest common divisor of N and M (referenced gcd(N,M)), with N and M being integers, is equal to 1 (N and M as prime numbers with respect to each other), then $N\Phi^{(M)}=1 \bmod(M)$ with $\Phi$ as the Euler function.

Euler's Theorem

Now, if $M=2n$, then $\Phi(M)=2^{n-1}$.

In the case of $GF(2^n)$ which is relevant here, assuming therefore that N is encoded on n bits:

either N is an even number and $gcd(N,M)=2^r$, with r as an integer smaller than or equal to n,

or N is an odd number and gcd(N,M)=1 and according to Euler's theorem, $N^D=1 \bmod(2^n)$ with $D=2^{n-1}$.

Thus, if N is an odd number, we have

$N^{-1}=N^E \bmod(2^n)$, with $E=D-1=2^{n-1}-1$.

The modular inversion therefore amounts to an exponentiation with a exponent E such that $l=k=n-1$, with l and k being defined identically to the manner described here above. The computation of the inverse of an odd number will therefore be done in a constant period of m cycles with:

$m=(2*n-3)*(n+1)$ if we take account of the steps 1 and 2 of the exponentiation as defined here above,

$m=2*(n-2)*(n+1)$ if the only steps taken into account are the other steps (the computation proper).

Two's Complement Operation

At present, the circuit ST16CF54 by SGS-THOMSON MICROELECTRONICS S.A. uses software methods to resolve the equation $J_0*N_0+1=0 \bmod(2^n)$, with n=32 and N as a known number encoded on n bits. This resolution, as indicated in the introduction, is implemented by the central processing unit of this circuit in an average time of 250 microseconds. During this time, its mathematical coprocessor is stopped. The program used to resolve this equation furthermore takes up 124 bytes of read-only memory.

It will be observed that the resolution of this equation is equivalent to resolving the equation $J_0=-N_0^{-1} \bmod(2^{32})$.

The circuit 11 as described here above enables the computation of $N_0^{-1} \bmod(2^{32})$. If we consider a clock signal with a frequency of 40 MegaHertz (the case of the ST16CF54), this computation requires, apart from the steps 1 and 2, a computation time of 49.5 microseconds and a total processing time (inclusive of the steps 1 and 2) of 50.3 microseconds. Thus, the period of resolution of the equation is reduced by a factor of 5 as compared with the ST16CF54. Given the structure of the circuit 11, which is of a wired logic type, this circuit can be easily integrated into an integrated circuit such as the one designated. Similarly, this circuit could easily be integrated into an existing mathematical coprocessor without any radical modification, given the simplicity of the circuit described. Furthermore, the approach of the invention enables a gain in read-only memory since the resolving of the equation no longer mobilizes the resources of the central processing unit.

It will be noted that although the invention refers to a particular product, it is nonetheless usable in any other circuit whose application pertains to the above-defined operations in a Galois field.

In the context of the ST16CF54, it is sought to resolve the equation $J_0=-N_0^{-1} \bmod(2^{32})$. The circuit 11 as described here above enables the computation of $-J_0$. It is possible in

## Euler's theorem

Date: 7/2/96 at 13:46:32
From: Anonymous
Subject: Euler's Theorem

I am not sure how to find the inverse of a modulo m using Euler's theorem. Using the formula of my book, I end up just getting a. The answers I get are also not what the book has as the answers. Any hints would be appreciated.

---

Date: 7/2/96 at 17:30:44
From: Doctor Anthony
Subject: Re: Euler's Theorem

Euler's theorem states that if (a,m)=1 (i.e. a and m are relatively prime), then a^(phi(m)) == 1 (mod m) where phi(m) is the number of integers less than m and prime to it. → e.g., $a^{\Phi(m)} = 1 \pmod m$

Euler's Theorem

If phi(m) = n then a^n == 1 (mod m)

So a*a^(n-1) == 1 (mod m) but a*a^(-1) == 1 (mod m)

and so a^(-1) == a^(n-1) (mod m)

-Doctor Anthony, The Math Forum
Check out our web site! http://mathforum.org/dr.math/

**Associated Topics:**
High School Number Theory

---

**Submit your own question to Dr. Math**

[Privacy Policy] [Terms of Use]

---

Math Forum Home || Math Library || Quick Reference || Math Forum Search

---